活体检测接入文档

一、SDK集成

1、获取SDK

从github上下载活体检测sdk的aar包 点我下载sdk

2、手动导入SDK

将获取的sdk的aar文件放到工程中的libs文件夹下，然后在app的build.gradle文件中增加如下代码

```
repositories {
    flatDir {
        dirs 'libs'
    }
}
```

在dependencies依赖中增加对aar包的引用

```
implementation(name:'alive_detected_libary', ext: 'aar')//aar名称和版本号以下载下来的最新版为准
implementation(name: 'openCVLibrary343', ext: 'aar')    // 添加对OpenCV库的依赖
implementation 'com.squareup.okhttp3:okhttp:3.3.1'      // 添加对okHttp的依赖
implementation 'com.google.code.gson:gson:2.8.5'        // 添加对gson的依赖
```

二、SDK接口

1）活体检测功能提供类：AliveDetector

- getInstance()：获取AliveDetector单例对象
- init(Context context, NISCameraPreview cameraPreview, String businessId)：初始化，第一个参数是Context对象，第2个参数为相机预览View，第三个参数为从易盾官网申请的业务id
- void setDetectedListener(DetectedListener detectedListener)：设置回调监听器
- void startDetect()：开始检测
- void stopDetect()：关闭检测
- void setTimeOut(long timeout)：设置检测超时时间，单位ms，默认为2min

2）活体检测检回调监听器类：DetectedListener

```
public interface DetectedListener {
    /**
     * 活体检测引擎已经初始化完成，可以开始检测
     */
    void onReady();
```

```
    /**
     * 活体检测状态是否改变，当引擎检测到状态改变时会回调该接口
     *
     * @param stateTip:引擎检测到的实时状态
     */
    void onStateTipChanged(String stateTip);

    /**
     * 活体检测是否通过回调
     */
    void onPassed(boolean isPassed,String token);

    /**
     * 活体检测过程中出现错误时回调
     *
     * @param code:错误码
     * @param msg:出错原因
     */
    void onError(int code, String msg,String token);

    /**
     * 活体检测过程超时回调
     */
    void onOverTime();
}
```

三、使用说明

1、在xml布局文件中使用活体检测相机预览View

如下是个简单示例：

```
 <com.netease.nis.alivedetected.NISCameraPreview
        android:id="@+id/surface_view"
        android:layout_width="match_parent"
        android:layout_height="400dp" />
```

1、获取AliveDetector对象，进行初始化

将前面布局中获取到的相机预览View以及从易盾官网申请的业务id传给init()接口进行初始化

```
mAliveDetector = AliveDetector.getInstance();
mAliveDetector.init(this, mCameraPreview, BUSINESS_ID);
```

2、设置回调监听器，在监听器中根据相应回调做自己的业务处理

```
 mAliveDetector.setDetectedListener(new DetectedListener() {
            @Override
            public void onReady() {
```

```
                //Toast.makeText(getApplicationContext(), "检测引擎初始化完成",
Toast.LENGTH_LONG).show();
            }

            @Override
            public void onStateTipChanged(String stateTip) {
                setTipText(stateTip);
            }

            @Override
            public void onPassed(boolean isPassed, String token) {
                if (isPassed) {
                    showToast("活体检测通过,token is:" + token);
                    // 可以在此处使用token做reCheck
                } else {
                    showToast("活体检测不通过,token is:" + token);
                }
            }

            @Override
            public void onError(int code, String msg, String token) {
                Log.e(TAG, "listener [onError]:" + msg);
                showToast("活体检测出错,原因:" + msg + " token:" + token);
            }

            @Override
            public void onOverTime() {
                showToast("检测超时");
            }

        });
```

3、开始/停止检测

```
mAliveDetector.startDetect();
mAliveDetector.stopDetect();
```